НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп‘ютерних систем

«До захисту допущено»

Завідувач кафедри

_____ Тарасенко В.П.
(підпис) (ініціали, прізвище)

“___” червня 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напряму підготовки **6.050102 «Комп'ютерна інженерія»**

на тему: “Мобільний додаток для управління пристроями «Розумного дому»”.

Виконав: студент IV курсу, групи КВ-51

Микитенко Сергій Сергійович _____
(підпис)

Керівник, асистент каф. СПіСКС, Щербина Б.О. _____
(підпис)

Консультант з нормоконтролю, доц.каф.СПіСКС, к.т.н. Клятченко Я.М. _____
(підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2019 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп‘ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Напрям підготовки 6.050102 «Комп’ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Тарасенко В.П.
(підпис) (ініціали, прізвище)
«___»червня 2019 р.

ЗАВДАННЯ
на дипломний проект студента
Микитенко Сергія Сергійовича

1. Тема проекту: “Мобільний додаток для управління пристроями «Розумного дому»”.

Керівник проекту: асистент каф. СПіСКС, Щербина Б.О,

затверджені наказом по університету від «06»квітня 2018 р. №1108-С

2. Термін подання студентом проекту _____

3. Вихідні дані до проекту:див. технічне завдання.

4. Зміст пояснювальної записки:аналіз існуючих рішень та обґрунтування теми диплому, комунікація пристроїв системи «Розумний дім», бібліотека для реалізації інтерфейсу комунікації в пристроях системи «Розумний дім», мобільний додаток для управління пристроями системи «Розумний дім», тестування системи.

5. Перелік графічного матеріалу (із зазначенням обов’язкових креслеників, плакатів, презентацій тощо): структура бібліотеки інтерфейсу комунікації

пристроїв системи «Розумний дім», алгоритм обробки запитів від мобільного додатку пристроєм системи «Розумний дім», шаблони елементів «button», «trigger», «sensor», «selector», структура мобільного додатку для управління пристроями системи «Розумний дім».

6. Консультанти розділів проекту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М., доц. каф. СПіСКС, к.т.н.		

7. Дата видачі завдання_____

Календарний план

№ з/п	Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання	Термін виконання
1.	Видача завдання на дипломне проектування	04.11.2018
2.	Вивчення літератури за тематикою роботи	16.11.2018
3.	Розроблення та узгодження технічного завдання	25.11.2018
4.	Розроблення структури додатку	17.01.2019
5.	Розроблення дизайну та графічних елементів	04.02.2019
6.	Програмна реалізація додатку	15.03.2019
7.	Тестування додатку	04.04.2019
8.	Підготовка матеріалів текстової частини проекту	24.04.2019
9.	Підготовка матеріалів графічної частини проекту	17.05.2019
10.	Оформлення технічної документації проекту	28.05.2019

Студент _____ ___Микитенко С.С.____
(підпис) (ініціали, прізвище)

Керівник проекту _____ _____Щербина Б.О.____
(підпис) (ініціали, прізвище)

*Консультантом не може бути зазначено керівника дипломного проекту.

АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (51 с., 32 рис., 4 додатки).

Бакалаврський проект призначено розробці мобільного додатку управління пристроями «Розумного дому», який дозволяє вести моніторинг і управління пристроями через домашню локальну мережу.

Мобільний додаток дозволяє: встановлювати з'єднання з пристроями в локальній мережі; здійснювати передачу даних бездротовими каналами зв'язку; управляти пристроями «Розумного дому» за допомогою спеціальних команд. Передбачена можливість одночасного підключення декількох пристроїв. В процесі розробки були використані технології бездротового зв'язку Wi-Fi 802.11 b/g/n. В якості апаратної бази пристроїв «Розумного дому» використовувався мікроконтролер ESP8266.

В ході розробки:

- проведено аналіз існуючих систем «Розумного дому»;
- сформульовані вимоги до мобільного додатку управління пристроями системи «Розумного дому»;
- розроблена система комунікації пристроїв системи «Розумного дому»;
- розроблено програмне забезпечення мікроконтролера для прийому і виконання команд управління;
- розроблено мобільний додаток для управління пристроями системи «Розумного дому».

Використання програмних засобів наведених в цій роботі дозволяє розробникам зменшити час затрачений на створення пристроїв систем «Розумного дому».

Ключові слова:

МОБІЛЬНИЙ ДОДАТОК ДЛЯ УПРАВЛІННЯ ПРИСТРОЯМИ «РОЗУМНОГО ДОМУ», WIFI 802.11 B/G/N, ESP8266, C/C++, ANDROID, JAVA.

ABSTRACT

The diploma project includes an explanatory note (51 p., 32 fig., 4 appendices).

The Bachelor's project is designed to develop a mobile application for managing Smart Home devices, which performs monitoring and management of the devices through the home network.

The mobile application can perform next actions : connect to devices on a local network; transmit data via wireless communication channels; manage Smart Home devices with special commands. Also, the capability of simultaneous connection of several devices was provided. In the development process were used wireless technologies, namely Wi-Fi 802.11 b/g/n. As a hardware base of Smart Home devices, was used the ESP8266 microcontroller.

In the development process was:

- analyzed existing Smart Home systems;
- formulated requirements for the mobile application for managing Smart Home devices;
- developed communication system for the Smart Home devices;
- developed microcontroller software for reception and execution of control commands;
- developed the mobile application for controlling devices of the Smart Home system.

Using the software provided in this work allows developers to reduce the time spent on creating their Smart Home systems and devices.

Key words:

MOBILE APPLICATION FOR MANAGEMENT OF SMART HOME DEVICES, WIFI 802.11 B/G/N, ESP8266, C/C++, ANDROID, JAVA.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	А4	ІАЛЦ.045492.002 ТЗ	Мобільний додаток для	4		
			управління пристроями			
			системи «Розумний дім»			
			Технічне завдання			
	А4	ІАЛЦ.045492.003 ТП	Мобільний додаток для	2		
			управління пристроями			
			системи «Розумний дім»			
			Відомість технічного			
			проекту			
	А4	ІАЛЦ.045492.004 ПЗ	Мобільний додаток для	51		
			управління пристроями			
			системи «Розумний дім»			
			Пояснювальна записка			
	А4	ІАЛЦ.045492.005 Е1	Структура бібліотеки для	1		
			реалізації інтерфейсу			
			пристроїв системи			
			«Розумний дім».			
			Схема структурна			

Змін.	Арк.	№ докум.	Підпис	Дата	ІАЛЦ.045492.001 ОА		
Розробив		Микитенко С.С.			Мобільний додаток для управління пристроями «Розумного дому» **Опис альбому**	Літ.	Аркуш 1 / Аркушів 2
Перевірив		Щербина Б.О.					
Консульт.						КПІ ім. Ігоря Сікорського, ФПМ КВ-51	
Н. контроль		Клятченко Я.М.					
Зав. каф.		Тарасенко В.П.					

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	А4	ІАЛЦ.045492.006 Д2	Алгоритм обробки запитів	1		
			від мобільного додатку			
			пристроєм системи			
			«Розумний дім».			
			Схема алгоритму			
	А4	ІАЛЦ.045492.007 Д3	Шаблони елементів	1		
			«button», «trigger», «sensor»			
			«selector».			
			Діаграми класів			
	А4	ІАЛЦ.045492.008 Д1	Структура проекту	1		
			мобільного додатку для			
			управління пристроями			
			системи «Розумний дім».			
			Схема структурна			
		Диск CD-ROM	Текст пояснювальної	1		
			записки.			
			Графічний матеріал			

ЗМІСТ

					ІАЛЦ. 467200.002 ТЗ		
Змін	Арк.	№ докум.	Підпис	Дата			
Розробив		Микитенко С.С.			*Мобільний додаток для управління пристроями «Розумного дому»*	Літ.	Аркуш 1 / Аркушів 4
Перевірив		Щербина Б.О.					
Н. контроль		Клятченко Я.М.				КПІ ім. Ігоря Сікорського, ФПМ КВ-51	
Затвердив		Тарасенко В.П.			***Технічне завдання***		

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Мобільний додаток управління пристроями «Розумного дому»».

Галузь застосування: створення систем «Розумного дому», що керуються через один мобільний додаток.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є створення мобільного клієнта під операційну систему Android для управління та моніторингу систем «розумного дому».

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації в періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- сумісність з операційною системою Android;

- можливість підключення пристроїв розумного дому;
- можливість управління пристроями розумного дому;
- можливість отримання інформації від пристроїв розумного дому;
- можливість відключення пристроїв домашнього розуму;
- наявність зручної системи вибору команди управління;
- прив'язка ip та порта до пристроя розумного дому;
- зберігання встановленого зв’язку з пристроем розумного дому.

5.2. Вимоги до апаратного забезпечення

- оперативна пам’ять: 1 Гб;
- наявність доступу до мережі Wi-Fi (IEEE 802.11 b/g/n).

5.3. Вимоги до програмного та апаратного забезпечення користувача

- операційна система Android;
- наявність доступу до мережі Wi-Fi (IEEE 802.11 b/g/n).

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Видача завдання на дипломне проектування	04.11.2018
2.	Вивчення літератури за тематикою роботи	16.11.2018
3.	Розроблення та узгодження технічного завдання	25.11.2018
4.	Розроблення структури додатку	17.01.2019
5.	Розроблення дизайну та графічних елементів	04.02.2019
6.	Програмна реалізація додатку	15.03.2019
7.	Тестування додатку	04.04.2019
8.	Підготовка матеріалів текстової частини проекту	24.04.2019
9.	Підготовка матеріалів графічної частини проекту	17.05.2019
10.	Оформлення технічної документації проекту	28.05.2019

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	А4	ІАЛЦ.045492.004 ПЗ	Мобільний додаток для	51		
			управління пристроями			
			системи «Розумний дім»			
			Пояснювальна записка			
	А4	ІАЛЦ.045492.005 Е1	Структура бібліотеки для	1		
			реалізації інтерфейсу			
			пристроїв системи			
			«Розумний дім».			
			Схема структурна			
	А4	ІАЛЦ.045492.006 Д2	Алгоритм обробки запитів	1		
			від мобільного додатку			
			пристроєм системи			
			«Розумний дім».			
			Схема алгоритму			
	А4	ІАЛЦ.045492.007 Д3	Шаблони елементів	1		
			«button», «trigger», «sensor»			
			«selector».			
			Діаграми класів			

Змін.	Арк.	№ докум.	Підпис	Дата	ІАЛЦ.045492.003 ТП		
Розробив		Микитенко С.С.			Мобільний додаток для управління пристроями «Розумного дому» **Відомість технічного проекту**	Літ.	Аркуш 1 / Аркушів 2
Перевірив		Щербина Б.О.					
Консульт.						КПІ ім. Ігоря Сікорського, ФПМ КВ-51	
Н. контроль		Клятченко Я.М.					
Зав. каф.		Тарасенко В.П.					

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	А4	ІАЛЦ.045492.008 Д1	Структура проекту	1		
			мобільного додатку для			
			управління пристроями			
			системи «Розумний дім».			
			Схема структурна			
		Диск CD-ROM	Текст пояснювальної	1		
			записки.			
			Графічний матеріал			

Змін.	Арк.	№ докум.	Підпис	Дата	ДП.468300.003 ТП	Арк. 2

ЗМІСТ

Додаток 1. Копії графічних матеріалів

- ІАЛЦ.045492.005 Е1. Структура бібліотеки інтерфейсу комунікації пристроїв системи «Розумний дім». Схема структурна
- ІАЛЦ.045492.006 Д2. Алгоритм обробки запитів від мобільного додатку пристроєм системи «Розумний дім». Схема алгоритму
- ІАЛЦ.045492.007 Д3. Шаблони елементів «button», «trigger», «sensor», «selector». Діаграми класів
- ІАЛЦ.045492.008 Е1. Структура мобільного додатку для управління пристроями системи «Розумний дім». Схема структурна

Додаток 2. Лістинг програми

- Приклад програми мікроконтролера пристроя системи «Розумного дому».

Додаток 3. Презентація

Додаток 4. Публікація за темою роботи

- ІПРТК-2019 «Підхід до комунікації пристроїв системи розумного дому».

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

API	Application Programming Interface	інтерфейс прикладного програмування
HTTP	Hyper Text Transfer Protocol	протокол передачі гіпер-текстових документів
IDE	Integrated Development Environment	інтегроване середовище розробки
JSON	JavaScript Object Notation	текстовий формат обміну даними між комп'ютерами
OSP	Open Source Project	проект з відкритим вихідним кодом
UART	Universal Asynchronous Receiver/Transmitter	універсальний асинхронний приймач/передавач
WEB	World Wide Web	міжнародна інформаційна система сховища інформації в електронному вигляді
WEB-інтерфейс	Веб-інтерфейс	сукупність засобів, за допомогою яких користувач взаємодіє з веб-сайтом або веб-застосунком через браузер
Голосовий помічник	Voice Assistant	це програмний агент, що може надавати персональну інформацію, виконувати завдання та послуги для окремої особи за допомогою голосових або текстових команд

Інтернет	Internet	всесвітня система сполучених комп'ютерних мереж
Мікроконтролер	Microcontroller	однокристальний комп'ютер, здатний виконувати прості завдання
Мобільний додаток	Mobile app	програмне забезпечення, призначене для роботи на смартфонах, планшетах та інших мобільних пристроях
хедер	header	Файл – заголовок програми мовою програмування C/C++

ВСТУП

На сьогодні існує велика різноманітність систем розумного дому. Більшість систем розумного дому розповсюджуються на комерційній основі, проте це не відміняє того факту, що будь-яка людина з хоча б мінімальним технічним досвідом може знайти в Інтернеті бажаний проект з відкритим кодом (OSP) і на його основі спробувати створити свій власний пристрій чи навіть цілу систему розумного дому.

Проблема виникає при спробі інтеграції одного пристрою в уже наявну систему, що побудована на основі пристроїв іншого виробника чи розробника у випадку OSP пристроїв. Найбільша проблема виникає з управлінням новим пристроєм, оскільки часто пристроями одного виробника не можна управляти засобами іншого.

Для вирішення проблеми сумісності необхідно створити єдиний алгоритм функціонування пристроїв розумного дому, за допомогою якого користувачі зможуть абстрагуватися від виробника чи розробника системи розумного дому і використовувати засоби керування незалежно від їх розробника.

В даному проекті розглянуто процес створення мобільного додатку для операційної системи Android, що дозволяє управляти пристроями розумного дому на базі мікросхем ESP8266. Розроблена система використовує спеціальну бібліотеку, що стандартизує комунікацію між додатком та пристроями системи «Розумний дім».

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

1.1. Аналіз існуючих комерційних систем «Розумного дому»

На даний момент існує значна кількість пристроїв розумного дому, що може охопити майже будь-які сценарії щоденного життя типової людини. Здебільшого це пристрої невідомих брендів, що не відрізняються особливою зручністю для користувачів, проте існують і пристрої відомих брендів (Amazon, Google, Xiaomi), які здебільшого мають більш зручний інтерфейс керування і знижену кількість помилок в порівнянні з невідомими пристроями невідомих брендів. Загалом всі ці пристрої системи розумного дому відносяться до однієї чи декількох з наступних груп:

1) пристрої, що керуються з мобільного додатку;
2) пристрої, що керуються голосовим помічником (voice assistant);
3) пристрої, що є медіаторами при спілкуванні з голосовим помічником;
4) пристрої, що не потребують зовнішні засоби управління;
5) пристрої, що керуються іншими пристроями;
6) пристрої, що беруть на себе управління іншими пристроями.

Більш відомі виробники зазвичай виробляють пристрої, які відносяться до 6 і 5 чи 2 і 3 групи, створюючи тим самим свої власні інфраструктури, що ускладнюють використання пристроїв інших виробників і тим самим змушуючи своїх користувачів продовжувати купувати пристрої свого сімейства [1-3].

Менш відомі виробники мають тенденцію розробляти пристрої 1 або 4 групи, тим самим не створюючи перешкод для своїх користувачів, що дозволяє їм нарощувати свою систему розумного дому пристроями інших

виробників. Але і в цьому випадку існує значний недолік, який полягає в тому, що збільшуючи кількість таких пристроїв користувач збільшує необхідну кількість мобільних застосунків для їх керування, навіть якщо він добавляє в свою систему розумного дому декілька пристроїв одного виробника, але різних типів.

Якщо розглядати картину ринку систем розумного дому в цілому, то більш правильно буде розділити всі пристрої на три великих сімейства:

- сімейство пристроїв, які створюються навколо голосового помічника;
- сімейство пристроїв, що керується мобільним додатком;
- сімейство пристроїв, що керується за допомогою шлюза, який в свою чергу керується за допомогою мобільного додатку.

До першого сімейства здебільшого відносяться бренди, що мають власного голосового помічника (Amazon, Google, etc.) [1-2], а до третього бренди, що націлені на бюджетного покупця (Xiaomi, etc.) [3]. В свою чергу друге сімейство базується навколо типізованих пристроїв, що мають більш конкретне використання (камери, освітлення, розетки, тощо).

Загалом комерційні системи розумного дому не мають прямих переваг над некомерційними, окрім можливості використання їх одразу після купівлі з урахуванням випадків, коли пристрій потребує додаткової купівлі шлюзу цього виробника.

За результатом проведеного аналізу сформульовано переваги і недоліки комерційних рішень для систем розумного дому.

Переваги:

- швидка побудова інфраструктури системи розумного дому;
- легкість налаштування системи розумного дому;
- оптимізоване використання енергії.

Недоліки:

- одноманітний дизайн у більшості виробників;
- обмежена кольорова гама (зазвичай, тільки чорні та білі кольори);
- завищена ціна, особливо на найпростіші пристрої;
- необхідність у використанні неймовірної кількості мобільних додатків для управління всіма пристроями за умовами створення системи розумного дому, що буде відповідати сценарію користування, при якому користувач зможе задовольнити найбільшу кількість своїх вимог;
- дані користувача, зазвичай, обробляються на серверах виробниках, що хоча й зменшує навантаження на пристрої користувача, проте може завдати шкоди його приватності;
- відсутність прямої сумісності між пристроями різних виробників.

1.2. Аналіз існуючих некомерційних систем «Розумного дому»

Розглядаючи та аналізуючи некомерційні проектів можна сказати, що існує значна кількість проектів з відкритим кодом (OSP), що охоплюють більшість сценаріїв, які можуть знадобитись користувачу. Однак, навіть у цьому випадку можливі ситуації, коли наявних проектів недостатньо або їх працездатність недостатня для використання на практиці, проте головної перевагою таких проектів є якраз те, що будь-який користувач має доступ до вихідного коду. Це не тільки дозволяє користувачу знати про все, що робить його «розумний» пристрій, тим самим не переживати про приховані дії, які міг приховати виробник, але й редагувати програму таким чином, щоб вона виконувала дії, які необхідні користувачу : тобто надає користувачу легкий спосіб розробки нових пристроїв. Звісно користувачі можуть зіткнутися зі значними складностями через недостатній досвід в цій сфері, проте навіть

користувач з мінімальними знаннями може досягнути бажаного результату завдяки добре розвиненій спільноті, яка не тільки не відштовхує, але й надихає на майбутні успіхи.

Загалом в некомерційній системі можна виділити дві категорії пристроїв:

1) пристрої, побудовані за допомогою конструктора;
2) пристрої, побудовані з вихідного коду.

До першої категорії відноситься проект Blynk [5]. Він має значну перевагу для користувачів з нульовим досвідом програмування, оскільки не потребує ніякого програмування – користувач може досягнути своєї мети за допомогою програми-конструктора, яка охоплює значну кількість базових сценаріїв використання пристроїв системи «розумний дім». Варто зауважити, що відсутність можливості програмувати ці пристрої також є і недоліком цього проекту, оскільки унеможливлює не тільки гнучке налаштування пристроїв, але й погіршує розуміння системи користувачем. Також до недоліків варто віднести, те що для кожного пристрою створюється окремий додаток, тим самим збільшуючи загальну кількість встановлених додатків для керування пристроями розумного дому до кількості пристроїв в системі. Проте цей недолік втрачає свою вагу для користувачів, які вирішили просто спробувати автоматизувати невелику кількість їхньої домашньої рутини, для таких користувачів цей варіант є ідеальним через простоту використання та низьку ціну, яка має тільки апаратну складову [5].

Для другої категорії не вдалось знайти який-небудь проект, що значно спрощує розробку цих самих пристроїв. В цьому випадку більшість проектів мають власно розроблені програми для управління пристроями, і тому користувачам необхідно вже мати мінімальний досвіт програмування для створення бажаних пристроїв на основі вже існуючих. Звісно ситуації

коли користувачу потрібно внести значні правки до програми дуже рідкі, що пояснюється використанням платформи Arduino [6]. Це призводить до значного зменшення навантаження на нових розробників, створюючи оптимально простий підхід до програмування мікроконтролерів, що започаткувало велику спільноту розробників і як результат – безліч проектів, які охоплюють більшість сценаріїв роботи розумного дому.

Інтерфейс управління пристроями розумного дому можна розділити на два типи.

1. WEB-інтерфейс.

2. Мобільний додаток.

При створенні системи, що складається лише з одного чи невеликої кількості простих пристроїв, доречно застосувати перший тип інтерфейсу [7]. При збільшенні кількості пристроїв чи їх складності розробка інтерфейсу першого типу сильно ускладнюється, а простота використання таких пристроїв сильно зменшується. В цьому випадку з’являється потреба у створенні мобільного додатку під одну з популярних операційних систем, як, наприклад, Android. В цьому випадку зростає не тільки зручність використання системи розумного дому, але й її гнучкість.

Загалом для некомерційних пристроїв системи розумного дому можна виділити наступні переваги та недоліки.

Переваги:

- зменшення вартості пристроїв до вартості їх компонентів, за рахунок відсутності комерційної складової;
- можливість досягти будь-якого рівня гнучкості в обмін на час розробки;
- відсутність пропрієтарного програмного забезпечення забезпечує впевненість користувача в безпечності його приватності;

- користувач обробляє дані напряму в пристрої без використання сторонніх серверів, тим самим зменшуючи небезпеку втрати приватності;
- дозволяє максимально кастомізувати систему розумного дому.

Недоліки:

- збільшення складності створення системи розумного дому;
- збільшення часу необхідного для створення системи розумного дому.

1.3. Обґрунтування теми дипломного проекту

Темою дипломного проекту було обрано створення мобільного додатку для системи розумного дому з такими необхідними перевагами:

- один мобільний додаток для всіх пристроїв системи розумного дому;
- максимально спрощене створення інтерфейсу для пристрою в мобільному додатку за допомогою використання створеної бібліотеки;
- збереження достатньої гнучкості, що наявна при самостійній розробці мобільного додатку розробником;
- комунікація з пристроями за допомогою локальної бездротової мережі.

Даний мобільний додаток може бути використаний для управління будь-яким пристроєм, що підтримує комунікацію за прикладниим програмним інтерфейсом (API), який представлено в другому розділі цього дипломного проекту.

Цей додаток дозволяє не тільки стандартизувати інтерфейс управління пристроями і поєднати їх в одному місці, але й значно пришвидшити їх розробку за допомогою зручного інтерфейсу розробки.

1.4. Обґрунтування вибору середовища розробки

Для створення пристроїв розумного дому була обрана популярна мікросхема esp8266, що має достатньо розвинену спільноту розробників і тим самим буде найбільш дружньою для майбутніх користувачів розробленого пристрою. Для уникнення складностей, що можуть виникнути у майбутніх користувачів, середовищем розробки бібліотеки було обрано платформу Arduino [6].

Мобільний додаток було розроблено під операційну систему Android, на яку припадає 74.85% ринку мобільних операційних систем у 2018 році [8].

Розробка для згаданих вище середовищ дозволяє створити достатньо гнучку систему, що може бути легко модифікована в майбутньому.

2. КОМУНІКАЦІЯ ПРИСТРОЇВ СИСТЕМИ «РОЗУМНИЙ ДІМ»

2.1. Інтерфейс комунікації пристроїв системи «Розумний дім»

Перш ніж приступити до розробки системи «Розумний дім» необхідно визначитись з інтерфейсом прикладного програмування (API) [9]. Не зважаючи на значну кількість вже існуючих інтерфейсів, було прийнято рішення розробити для цього проекту власний інтерфейс, що зможе забезпечити максимальну гнучкість компонентів системи «Розумний дім».

Розроблений інтерфейс побудовано на базі протоколу HTTP (Hyper Text Transfer Protocol) [10]. Для визначення типів елементів пристроїв розумного дому було введено 4 функціональні класи елементів і 4 дії над ними.

До класів належать:

- button – виконує одну функцію за запитом дії *on;*
- trigger – виконує дві функції за запитом дії *on* і *off*;
- sensor – виконує одну функцію за запитом дії *get* і відправляє значення повернене функцією клієнту;
- selector – виконує одну функцію за запитом дії put, яка отримує з тіла запиту вміст поля json [11] «data» в якості аргументу.

Наведенні класи можуть комбінуватися для досягнення більш складних структур. Наприклад, для розробки термометра оточуючого середовища можна скомбінувати класи sensor і selector, це дозволить не тільки отримувати від пристрою дані температури, але й визначати, в якому форматі пристрій буде їх відправляти [12].

Елементи пристроїв системи розумного дому мають не тільки назву класів, але й унікальний ідентифікатор елементу класу. Це дозволяє не

тільки однозначно визначати, до якого елементу адресовано запит користувача, але й спрощує розуміння своєї програми розробнику, оскільки він може дати однакові назви елементам одного модуля (наприклад, в модулі термометра програміст може мати одночасно елементи сенсор і селектор з назвою термометр).

2.2. Формат запиту до пристроїв системи «Розумний дім»

На рисунках 2.1 – 2.4 представлені приклади запитів до пристроїв системи «Розумний дім». Метод запиту завжди визначено як GET, за яким в тому ж самому рядку наводиться шлях, що складається з назв класу, елементу і операції. Серед заголовків обов'язковими є тільки Content-Length і Content-Type для елементів класу selector; для всіх інших класів обов'язкові заголовки відсутні. При передачі даних використовується формат даних json (JavaScript Object Notation).

На рисунку 2.1 показано приклад запиту на виконання операції «on» елементом «buttonName» класу «button».

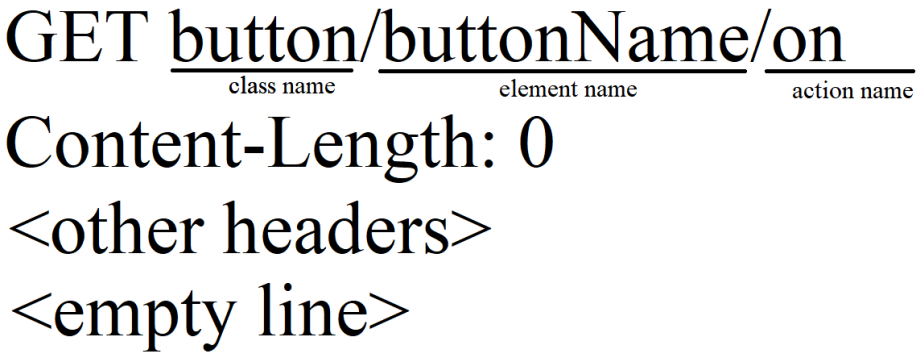

Рисунок 2.1 – Запит до елемента «buttonName» класу «button» на виконання операції «on»

На рисунку 2.2 показано приклад запиту на виконання операції «on» елементом «triggerName» класу «trigger».

GET trigger/triggerName/on
(trigger: class name; triggerName: element name; on: action name)
Content-Length: 0
<other headers>
<empty line>

Рисунок 2.2 – Запит до елемента «triggerName» класу «trigger» на виконання операції «on»

На рисунку 2.3 показано приклад запиту на виконання операції «get» елементом «sensorName» класу «sensor».

GET sensor/sensorName/get
(sensor: class name; sensorName: element name; get: action name)
Content-Length: 0
<other headers>
<empty line>

Рисунок 2.3 – Запит до елемента «sensorName» класу «sensor» на виконання операції «get»

На рисунку 2.4 показано приклад запиту на виконання операції «put» елементом «selectorName» класу «selector». Зауважимо, що функції put буде передано рядок «selectedString» в якості аргумента.

```
GET selector/selectorName/put
    class name   element name  action name
Content-Length: 31
Content-type: text/json
<other headers>
<empty line>
{
   "data":"selectedString"
}
```

Рисунок 2.4 – Запит до елемента «selectorName» класу «selector» на виконання операції «put»

Структура запиту дозволяє легко нарощувати кількість класів (типів) елементів та кількость інформації, що передається від клієнта (мобільний додаток) до сервера (пристрій системи «Розумний дім») за допомогою використання додаткових заголовків та додаткових полів файлу json (JavaScript Object Notation).

2.3. Формат відповіді пристрою системи «Розумний дім»

На рисунках 2.5 – 2.7 представлено приклади відповідей пристроїв системи «Розумний дім». Формат відповіді є стандартним для HTTP. Серед заголовків обов’язковими є тільки Content-Length і Content-Type для елементів класу sensor; для всіх інших класів обов’язкові заголовки відсутні. При передачі даних використовується формат даних json.

На рисунку 2.5 показано приклад відповіді в разі успішного виконання запиту до всіх класів, окрім «sensor». Код відповіді може відрізнятися.

```
HTTP/1.1 200 OK
Content-length: 0
<empty-line>
<empty-line>
```

Рисунок 2.5 – Відповідь в разі успішного виконання запиту до елементів класів «button», «trigger» та «selector»

На рисунку 2.6 показано приклад відповіді в разі помилки в запиті чи при виконанні. Код відповіді в цьому випадку обов’язково має бути більше 400, так як це впливає на функціонування мобільного додатку

```
HTTP/1.1 500 Internal Server Error
Content-length: 0
<empty-line>
<empty-line>
```

Рисунок 2.6 – Відповідь в разі якщо виконати запит не вдалось

На рисунку 2.7 представлено приклад відповіді в разі успішного виконання запиту до елементів класу «sensor». Поле даних «data» містить рядок «sensorDataString», що було повернено функцією get класу «sensor».

```
HTTP/1.1  200 OK
Content-length: 34
Content-type: text/json
<empty-line>
<empty-line>
{
   "data":"sensorDataString"
}
```

Рисунок 2.7 – Відповідь в разі успішного виконання запиту до елементів класу «sensor»

Структура відповіді дозволяє легко нарощувати кількість інформації, що передається від клієнта (мобільний додаток) до сервера (пристрій системи «Розумний дім») за допомогою використання додаткових заголовків та додаткових полів файлу json (JavaScript Object Notation). Також ця структура дозволяє збільшувати режимів тестування працездатності пристроїв в системі «Розумного дому», шляхом введення додаткових кодів та повідомлень відповіді.

3. БІБЛІОТЕКА ДЛЯ РЕАЛІЗАЦІЇ ІНТЕРФЕЙСУ КОМУНІКАЦІЇ В ПРИСТРОЯХ СИСТЕМИ «РОЗУМНИЙ ДІМ»

3.1. Структура бібліотеки

Метою створення бібліотеки було надання розробникам з недостатньою кваліфікацією в сфері розробки мобільних додатків можливості спрощення процесу створення мобільного додатку, задля чого платформою розробки було обрано Arduino IDE (Integrated Development Environment) [13]. Стандартна структура бібліотеки для Arduino IDE складається з файлу library.properties і папки src. Всі файли бібліотеки повинні знаходитись в папці src і бути вказаними в файлі library.properties.

Структурна схема бібліотеки зображена на рисунку 3.1. Бібліотека складається з 3 пакетів і одного файлу заголовку в корінній директорії (src). Файл заголовку wl_fox.h містить класс wl_fox, що є вхідною точкою програми інтерфейсу і відповідає за виконання всіх функцій бібліотеки. Бібліотека складається з наступних пакетів:

- wlfox_actions – містить абстрактні класи, що відповідають за представлення операцій над елементами (on, off, get, put);
- wlfox_classes – містить класи, що відповідають за представлення класів елементів (button, trigger, sensor, selector);
- wlfox_classes – містить класи, що відповідають за виконання комунікації пристроїв (server, reader, writer).

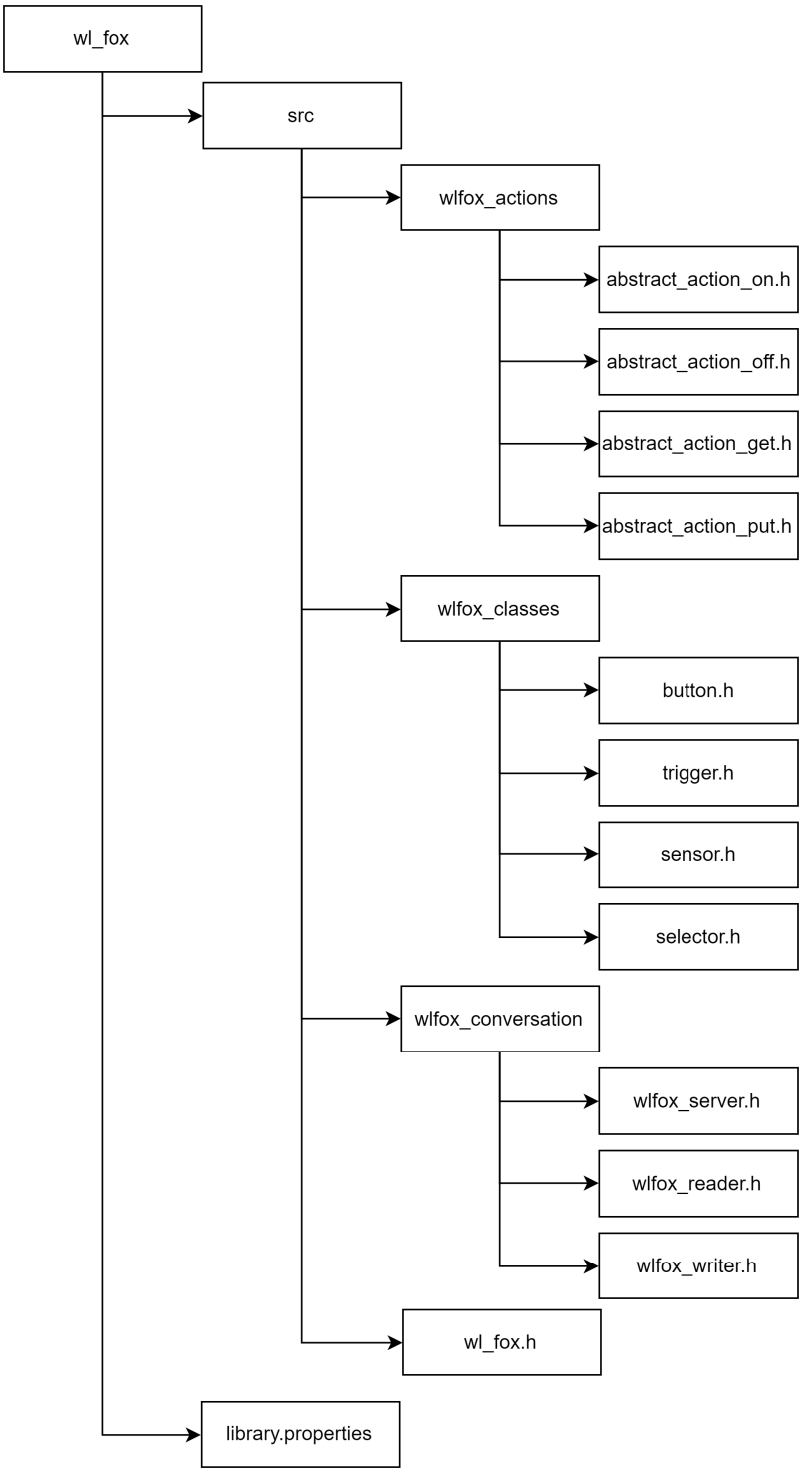

Рисунок 3.1 – Структурна схема бібліотеки

До пакету wlfox_actions входять наступні хедери(headers):

- abstract_action_on.h – містить абстрактний клас wlfox_on, що містить оголошення абстрактної функції void on();

- abstract_action_off.h – містить абстрактний клас wlfox_off, що містить оголошення абстрактної функції void off();
- abstract_action_put.h – містить абстрактний клас wlfox_put, що містить оголошення абстрактної функції void put(string arg);
- abstract_action_get.h – містить абстрактний клас wlfox_get, що містить оголошення абстрактної функції string get().

До пакету wlfox_classes входять наступні хедери:

- button.h – містить клас wlfox_button, що наслідує клас wlfox_on, реалізує функцію on та містить конструктор що приймає в якості аргументів посилання на функцію і унікальну назву елемента;
- trigger.h – містить клас wlfox_trigger, що наслідує класи wlfox_on та wlfox_off, реалізує функції on і off та містить конструктор що приймає в якості аргументів посилання на дві функції і унікальну назву елемента;
- sensor.h – містить клас wlfox_sensor, що наслідує клас wlfox_get, реалізує функцію get та містить конструктор що приймає в якості аргументів посилання на функцію і унікальну назву елемента;
- selector.h – містить клас wlfox_selector, що наслідує класи wlfox_get та wlfox_put, реалізує функції get і put та містить конструктор що приймає в якості аргументів посилання на дві функції і унікальну назву елемента.

До пакету wlfox_conversation входять наступні хедери:

- wlfox_server.h – містить клас wlfox_server, що реалізує методи комунікації з сокетами для мікросхеми esp8266, та має мету дозволити в майбутньому змінити мікросхему не переписуючи всю бібліотеку, а тільки один цей клас;

- wlfox_reader.h – містить клас wlfox_reader, що виконує читання та парсинг повідомлення, що надходять від клієнта;
- wlfox_writer.h – містить клас wlfox_writer, що виконує формування та надсилання результуючого повідомлення клієнту.

3.2. Методи керуючого класу

Бібліотека побудована навколо одного классу таким чином, щоб користувачу не доводилось імпортувати та використовувати більш ніж один класс бібліотеки. Головним хедером бібліотеки (тим, що має імпортувати розробник) є wl_fox.h, що мітить клас wl_fox.

Клас wl_fox містить наступні методи:

- wl_fox(int port) – конструктор, що ініціалізує клас і створює слухача для порту що був переданий в якості аргумента;
- void connect(const char* ssid, const char* password) – виконує підключення мікроконтролера до мережі wi-fi з заданим іменем і паролем;
- void startServer() – вмикає слухача, що був створений в конструкторі;
- void disconnect() – вимикає слухача, що був створений в конструкторі;
- void exec() – виконує ітерацію програми інтерфейса, перевіряє наявність підключеного клієнта і в разі його присутності обробляє його повідомлення;
- int newButton(String name, void(* func)()) – створює новий елемент класу button з унікальним іменем, що було передано в якості аргументу і встановлює передану функцію на виконання;
- int newTrigger(String name, void(* func1)() , void(* func2)()) – створює новий елемент класу trigger з унікальним іменем, що було

передано в якості аргументу і встановлює передані функції на виконання;

- int newSensor(String name, string(* func)()) – створює новий елемент класу sensor з унікальним іменем, що було передано в якості аргументу і встановлює передану функцію на виконання;
- int newSelector(String name, string(* func1)() , void(* func2)(string)) – створює новий елемент класу selector з унікальним іменем, що було передано в якості аргументу і встановлює передані функції на виконання.

3.3. Алгоритм роботи з бібліотекою

Робота з бібліотекою складається з п’яти кроків.

1. Створення екземпляру класу wl_fox.
2. Підключення пристрою до мережі wi-fi за допомогою виклику методу connect(ssid, password), де ssid – імя мережі, password - пароль до мережі ssid.
3. Створення елментів інтерфейсу шляхом передачі функцій та унікальних ідентифікаторів методам newButton, newTrigger, newSensor та newSelector.
4. Початок роботи серверу шляхом виклику метода startServer().
5. Виконання обробки клієнтів шляхом виклику методу exec() в безкінечному циклі.

В скетчі Arduino IDE кроки 1-4 мають виконуватись в функції setup(), а крок 5 – в функції loop().

4. МОБІЛЬНИЙ ДОДАТОК ДЛЯ УПРАВЛІННЯ ПРИСТРОЯМИ СИСТЕМИ «РОЗУМНИЙ ДІМ»

4.1. Структура мобільного додатку

Перед початком розробки була поставлена мета не тільки створити мобільний додаток під операційну систему, що охоплює більшу частину ринку мобільних операційних систем, але й використати таку мову програмування що дозволить в майбутньому мати мінімум труднощів з модифікуванням мобільного додатка. Такою операційною системою виявилась Android [8], а мовою програмування – Java [14].

Структурна схема програми мобільного додатку зображена на рисунку 3.1. Мобільний додаток створювався з урахуванням патерну MVC (Model View Controller) [15], за модель відповідає пакет java.ua.mykytenko.wl_fox.model, за контролер – пакет java.ua.mykytenko.wl_fox.controller, за вид – пакет res. Також в кореневій папці мобільного додатку знаходиться файл AndroidManifest.java, що містить важливу інформацію про програму мобільного додатку, яка потрібна операційній системі Android[16].

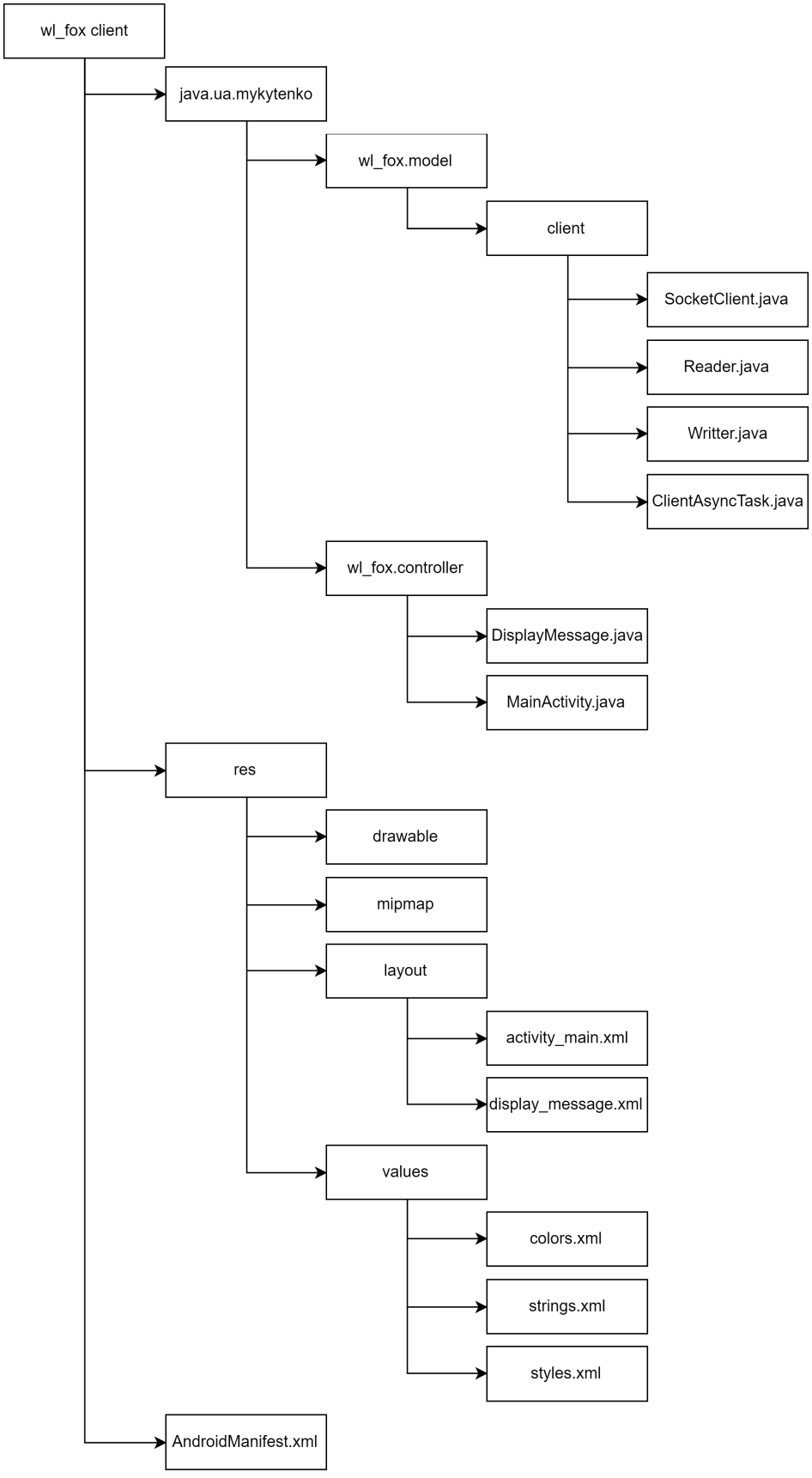

Рисунок 4.1 – Структурна схема мобільного додатку

Пакет java.ua.mykytenko.wl_fox.model містить наступні класи:

- java.ua.mykytenko.wl_fox.model.client.SocketClient.java – реалізує інтерфейс передачі даних через сокет від клієнта (мобільний додаток) до сервера (пристрою системи «Розумний дім»), при цьому він повністю абстрагований від даних, що передаються;
- java.ua.mykytenko.wl_fox.model.client.Reader.java – містить методи для читання та парсингу відповіді сервера (пристрою системи «Розумний дім»);
- java.ua.mykytenko.wl_fox.model.client.Writer.java – містить методи для підготовки та запису даних клієнту (мобільний додаток), що будуть передані на сервер (пристрій системи «Розумний дім»);
- java.ua.mykytenko.wl_fox.model.client.ClientAsyncTask.java – наслідує клас android.os.AsyncTask і реалізує метод doInBackground, що є необхідним для використання сокетів через особливості роботи операційної системи Android.

Пакет java.ua.mykytenko.wl_fox.controller містить наступні класи:

- java.ua.mykytenko.wl_fox.controller.MainActivity.java – наслідує клас android.app.Activity, що є точкою входу в програму та реалізує весь інтерфейс користувача, окрім відображення повідомлень про помилки;
- java.ua.mykytenko.wl_fox.controller.DisplayMessage.java – містить методи для відображення користувачу відповіді сервера в разі виникнення помилки.

Пакет res містить наступні файли:

- res.drawable.* – містить файли зображень;

- res.mipmap.* – містить файли зображень для значка мобільного додатку;
- res.drawable.activity_main.xml – містить макет інтерфейсу для класу java.ua.mykytenko.wl_fox.controller.MainActivity.java;
- res.drawable.display_message.xml – містить макет інтерфейсу для класу java.ua.mykytenko.wl_fox.controller.DisplayMessage.java;
- res.values.colors.xml – містить значення кольорів, що використовуються в додатку;
- res.values.strings.xml – містить значення рядків, що використовуються в додатку;
- res.values.styles.xml – містить значення стилів, що використовуються в додатку.

4.2. Опис алгоритму роботи додатку

При підготовці до роботи з мобільним додатком для управління пристроями «Розумного дому» користувачу необхідно виконати наступні кроки:

1) ввести ір адресу та порт пристрою системи «Розумний дім» у відповідні поля меню мобільного додатку;
2) дочекатися повідомлення про успішне підключення нового пристрою;
3) почекати поки додаток оновить інформацію про пристрій автоматично або натиснути кнопку «update» в блоці пристрою;
4) за необхідністю повторити кроки 1-3 для всіх пристроїв системи «Розумний дім»;
5) користуватися додатком, використовуючи кнопки інтерфейсу, що відображають елементи створених пристроїв системи «Розумний дім».

Алгоритм роботи мобільного додатку після натискання кнопки інтерфейсу користувачем складається з наступних кроків:

1) отримання параметрів з інтерфейсу користувача в класі java.ua.mykytenko.wl_fox.controller.MainActivity.java;
2) формування повідомлення пристрою системи «Розумний дім» за допомогою класу java.ua.mykytenko.wl_fox.model.client.Writer.java;
3) відправка повідомлення в фоні використовуючи клас java.ua.mykytenko.wl_fox.model.client.ClientAsyncTask.java;
4) очікування надходження відповіді;
5) читання відповіді використовуючи клас java.ua.mykytenko.wl_fox.model.client.ClientAsyncTask.java;
6) інтерпретування відповіді за допомогою класу java.ua.mykytenko.wl_fox.model.client.Reader.java;
7) якщо повідомлення було успішно відправлене і отримана відповідь від сервера, що вказує на успішне виконання операції, то оновлюються дані в інтерфейсі мобільного додатка за допомогою класу java.ua.mykytenko.wl_fox.controller.MainActivity.java, інакше формується повідомлення про помилку, що відображається за допомогою класу java.ua.mykytenko.wl_fox.controller.DisplayMessage.java.

4.3. Інтерфейс користувача

Інтерфейс користувача представляє собою вікно довільної довжини, що містить послідовно відображені інтерфейси всіх підключених пристроїв та в кінці – форму додавання нових пристроїв. Кожен з підключених пристроїв містить свою виділену форму на наступні поля і кнопки:

1) поле назви пристрою;
2) поле ip адреси пристрою;
3) поле порта пристрою;
4) кнопка оновлення інформації пристрою;
5) кнопка видалення пристрою.

Головним призначенням мобільного додатку є відображення інтерфейсу пристрою, що складається з наступних елементів:

1) button – відображається як кнопка з назвою всередині;
2) trigger – відображається як назва тригера з кнопкою напроти, що мітить в середині одне з двух значень – «on» і «off»;
3) sensor – відображається як назва сенсора з текстовим полем напроти, що містить інформацію, отриману від пристрою, а на другому рядку знаходиться кнопка оновлення цієї інформації;
4) selector – відображається як тестова форма з кнопкою «send» напроти.

5. ТЕСТУВАННЯ СИСТЕМИ

5.1. Підхід до тестування

Тестування може бути проведено за багатьма критеріями, наведемо деякі з розповсюджених:

1) виконання операцій за задовільний час;
2) відповідність вимогам проектувальника чи розробника;
3) правильна відповідь для усіх можливих наборів вхідних даних;
4) практичність та зручність користування.

В розробленому проекті можна не враховувати перший пункт, оскільки майже неможливо визначити задовільний час виконання програми для даного типу програм. На основі пунктів 2-4 розроблена програма тестування мобільного додатку для управління системою «Розумного дому», яка містить наступні кроки:

1) підключитись до пристроїв, що містять один з елементів (button, trigger, sensor, selector), перевірити правильність відображення елементів та виконання заданої задачі на стороні пристрою системи «Розумний дім»;
2) підключитись до декількох пристроїв одночасно, перевірити правильність функціонування мобільного додатку і пристроїв системи «Розумний дім»;
3) перезаписати програму пристрою, видаливши попередній елемент, спробувати звернутись до вже неіснуючого елементу з мобільного додатку, переконатись в появі повідомлення про помилку, оновити інформацію про пристрій в мобільному додатку, впевнитись в коректній роботі додатку і пристрою системи «Розумний дім»;
4) видалити підключений пристрій в додатку, впевнитись в коректному відображені інтерфейсу мобільним додатком;

5) створити пристрій з усіма можливими типами елементів, впевнитись в правильності роботи додатку і пристрою системи «Розумний дім»;
6) створити пристрій з двома елементами одного типу, впевнитись в правильності роботи додатку і пристрою системи «Розумний дім»;
7) впевнитись в появі повідомлення про помилку в разі спроби підключити неіснуючий пристрій (ip адреса або порт не відповідає існуючим пристроям в локальній мережі).

Для перевірки правильності роботу пристроїв системи «Розумний дім» було виконано логування через серійний порт (UART) [17] мікросхеми та стандартні засоби моніторингу Arduino IDE.

5.2. Тестування

На рисунку 5.1 показано інтерфейс мобільного додатку одразу після встановлення. Користувач має ввести адресу та порт пристрою, які він бажає підключити, після чого натиснути «connect». В разі успішного виконання операції користувач побачить інформацію нового пристрою, інакше появиться повідомлення про помилку.

Рисунок 5.1 – Інтерфейс мобільного додатку без підключених пристроїв

Для перевірки елемента «button» було написано тестову програму (рис. 5.2), на рисунку 5.3 показані зміни в інтерфейсі мобільного додатку після підключення пристрою, що містить елемент «button». Після натискання кнопки «ButtonName» пристрій успішно приймає та оброблює запит (рис. 5.4).

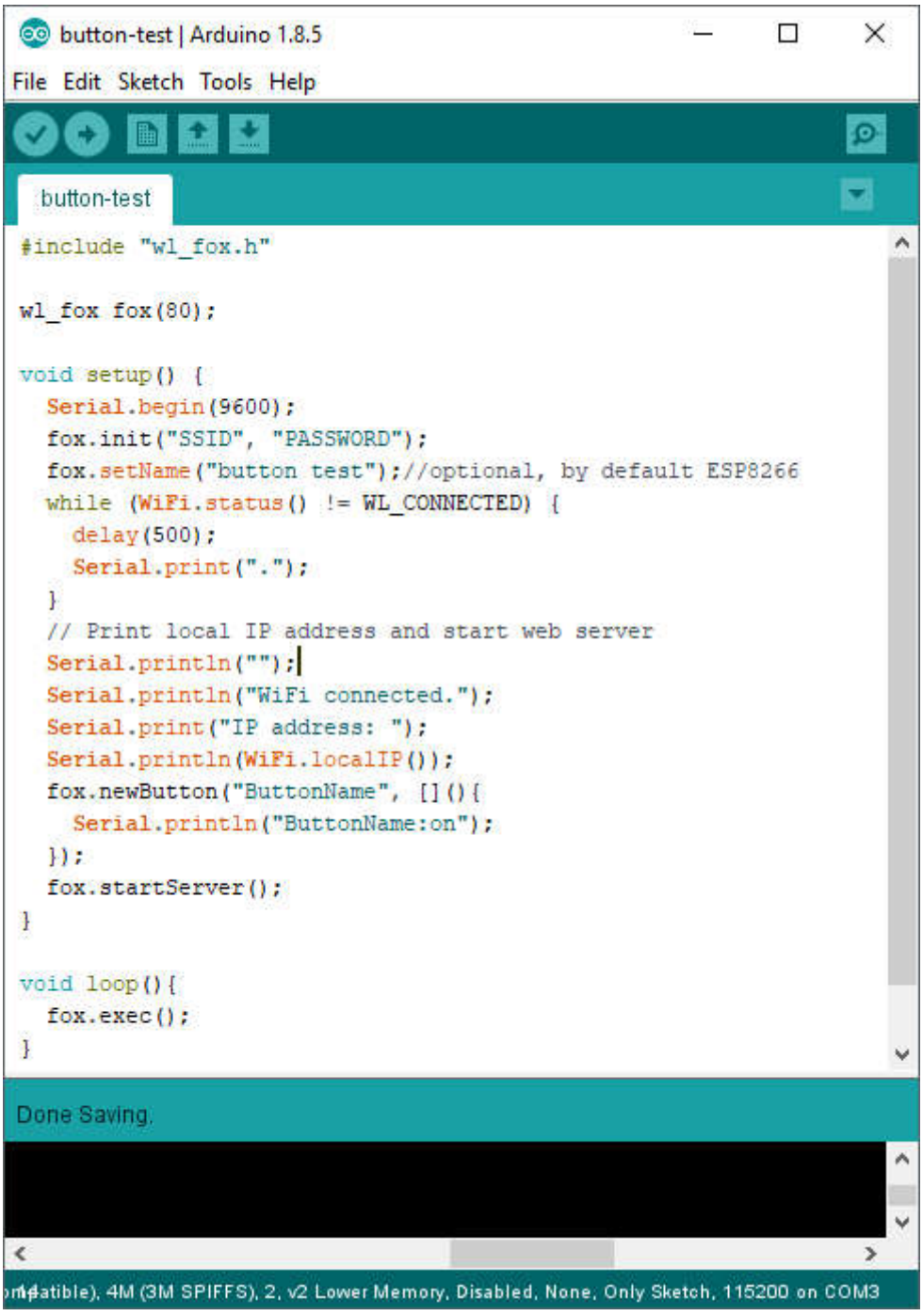

Рисунок 5.2 – Програма для тестування елемента типу «button»

Рисунок 5.3 – Інтерфейс мобільного додатку після підключення першого пристрою

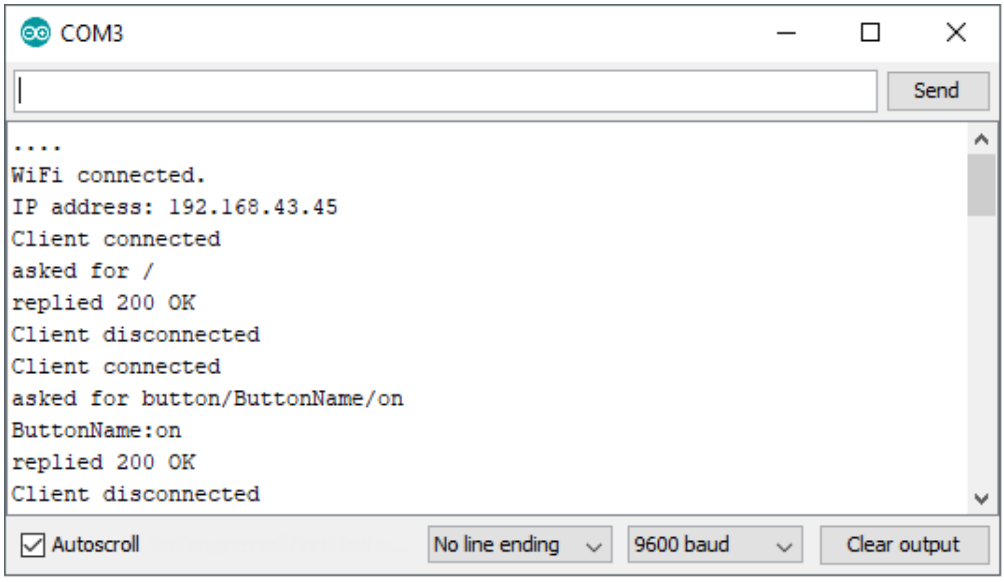

Рисунок 5.4 – Логи першого пристрою для тестування елемента типу «button»

Для перевірки елемента «trigger», а також можливості мобільного додатку підключати декілька пристроїв одночасно, було написано тестову програмну (рис. 5.5). На рисунку 5.6 показано зміни в інтерфейсі мобільного додатку після підключення пристрою, що містить елемент «trigger». Після натискання кнопки «on» пристрій успішно приймає та обробляє запит (рис. 5.7), а мобільний додаток змінює стан тригера на протилежний (рис. 5.8).

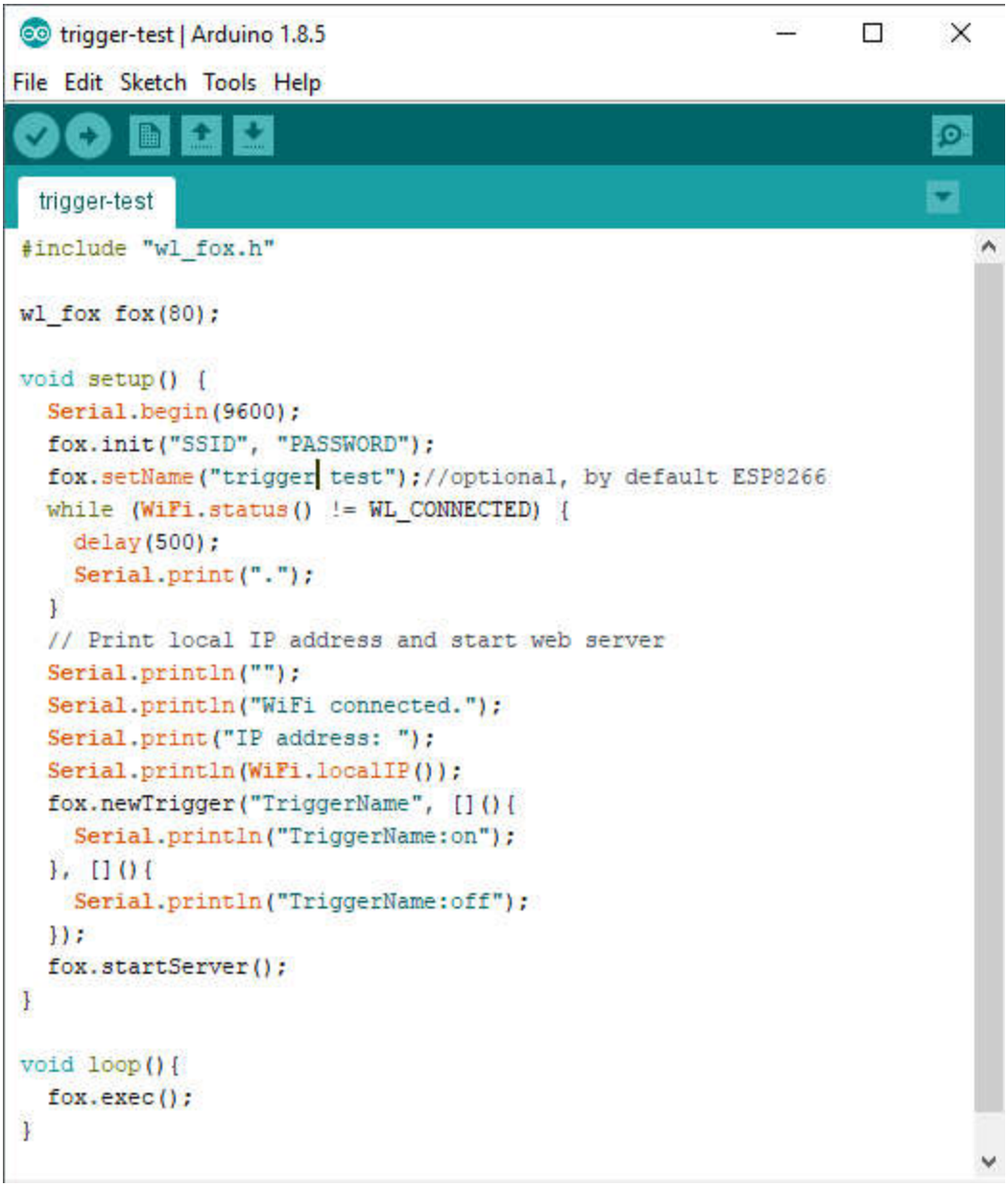

Рисунок 5.5 – Програма для тестування елемента типу «trigger»

Рисунок 5.6 – Інтерфейс мобільного додатку після підключення другого пристрою

Рисунок 5.7 – Інтерфейс мобільного додатку після виконання операції тригера

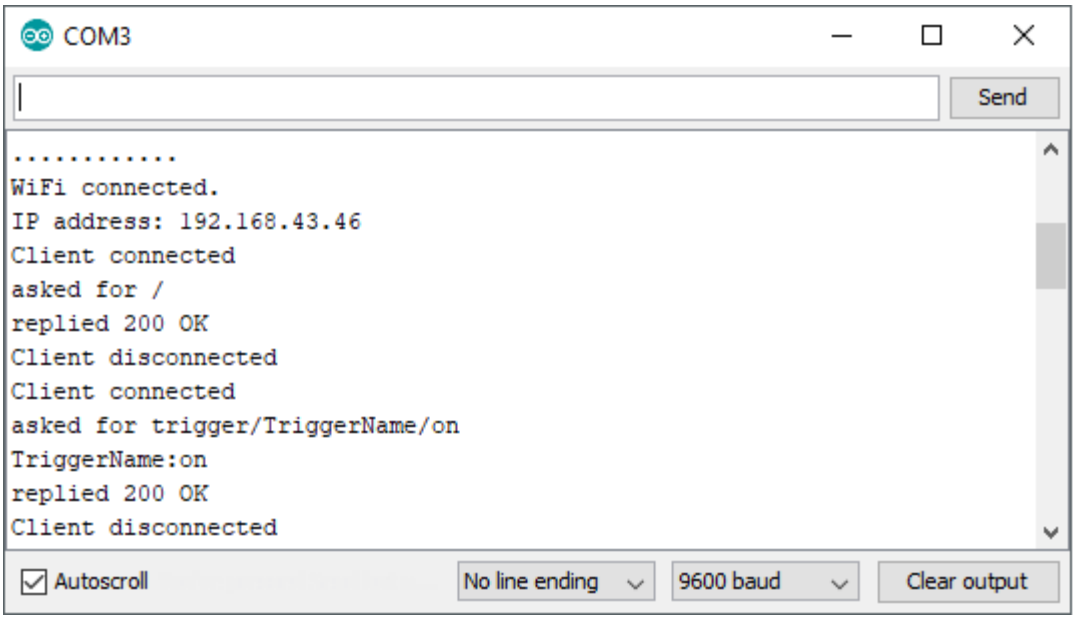

Рисунок 5.8 – Логи другого пристрою для тестування елемента типу «trigger»

Для перевірки елемента «sensor», а також правильності обробки помилок, було написано тестову программу (рис. 5.9) і перезаписано програму першого пристрою. Після натискання кнопки «ButtonName» пристрій успішно приймає та оброблює помилковий запит на вже неіснуючий елемент (рис. 5.10), а мобільний додаток змінює стан та формує повідомлення про помилку. Після натискання кнопки «update» в інтерфейсі першого пристрою його відображення успішно оновлюється і з’являється елемент «SensorName» (рис. 5.11). Натискання на кнопку «update» цього елемента успішно оброблється пристроєм (рис. 5.12).

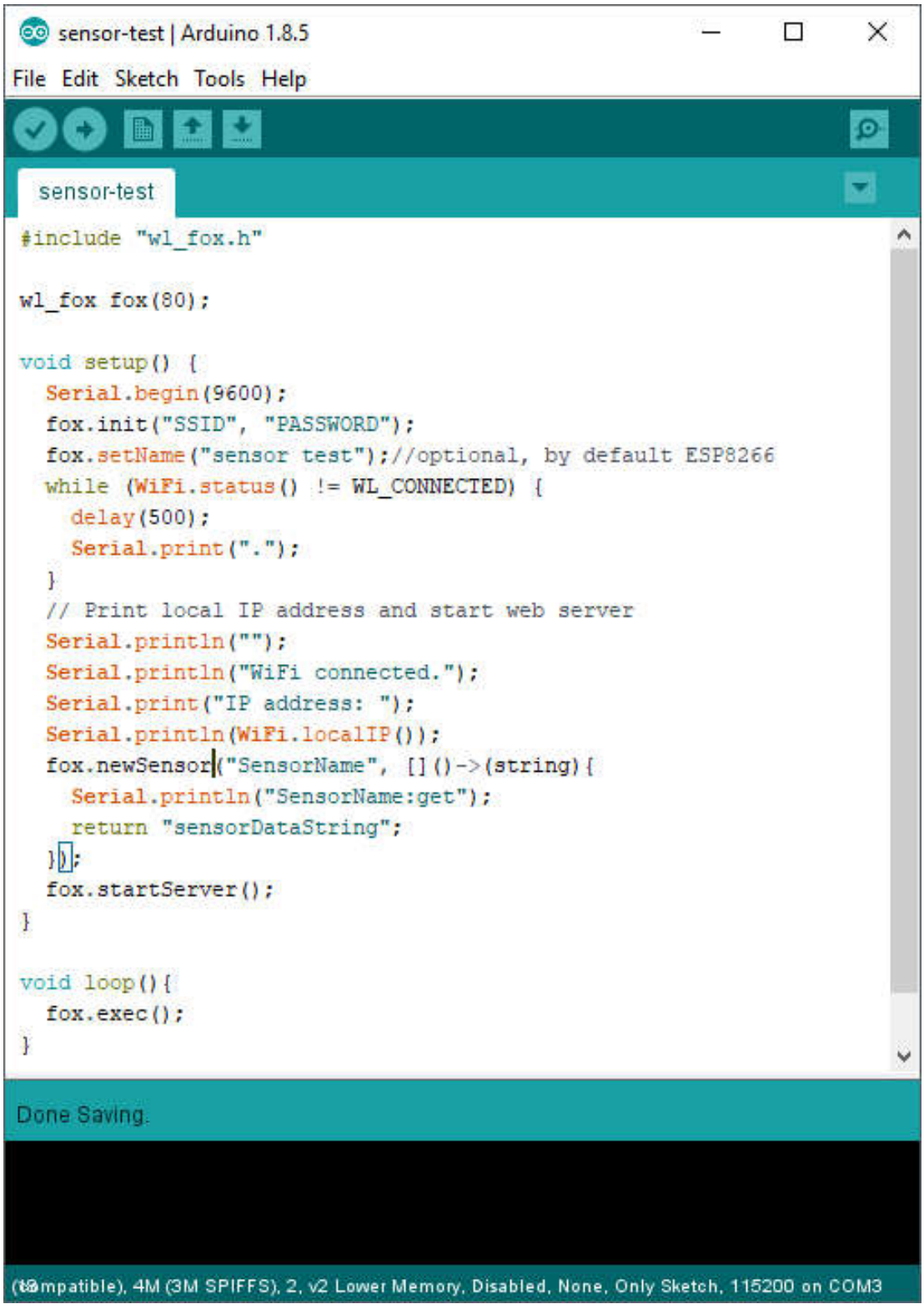

Рисунок 5.9 – Програма для тестування елемента типу «sensor»

Рисунок 5.10 – Інтерфейс мобільного додатку після помилкового запиту

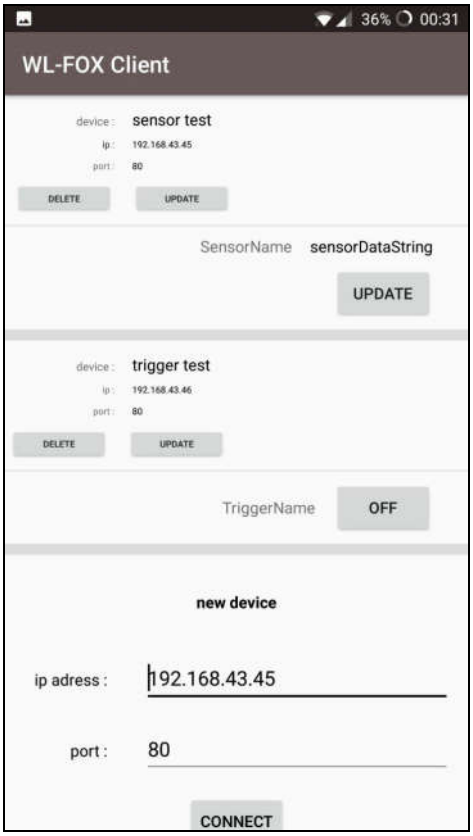

Рисунок 5.11 – Інтерфейс мобільного додатку після оновлення першого пристрою

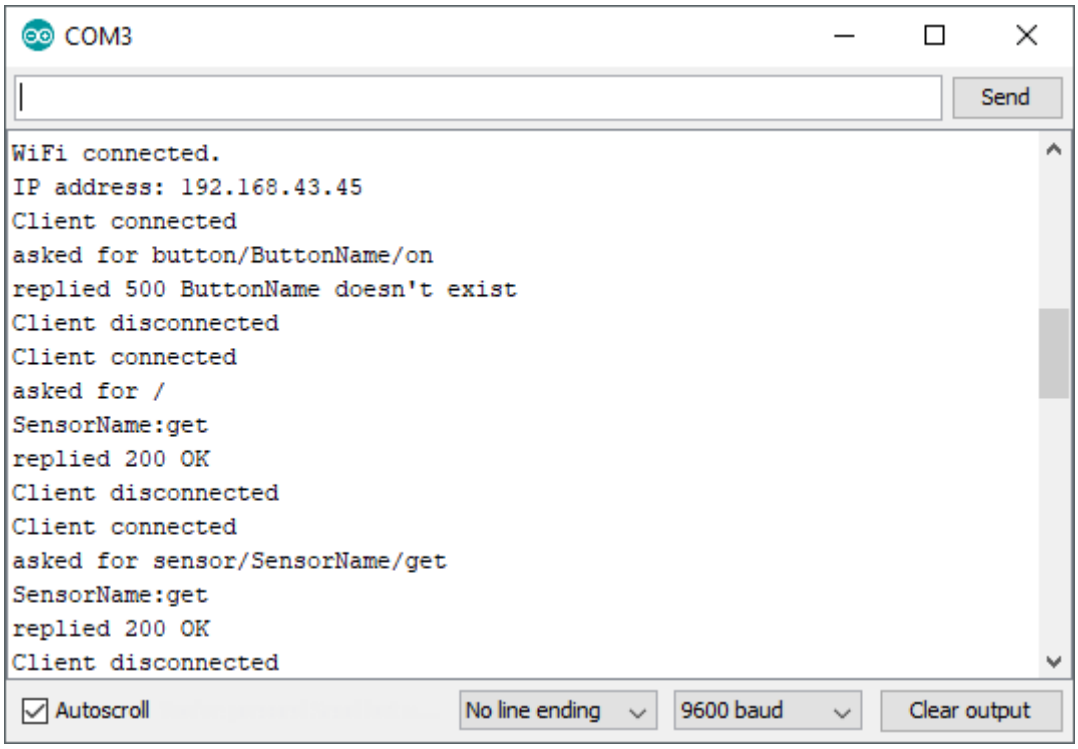

Рисунок 5.12 – Логи першого пристрою для тестування елемента типу «sensor»

Для перевірки елемента «selector» було написано тестову программу (рис. 5.13) і перезаписано програму другого пристрою. Після оновлення пристрою в мобільному додатку (рис. 5.14) пристрій успішно приймає та обробляє запит (рис. 5.15).

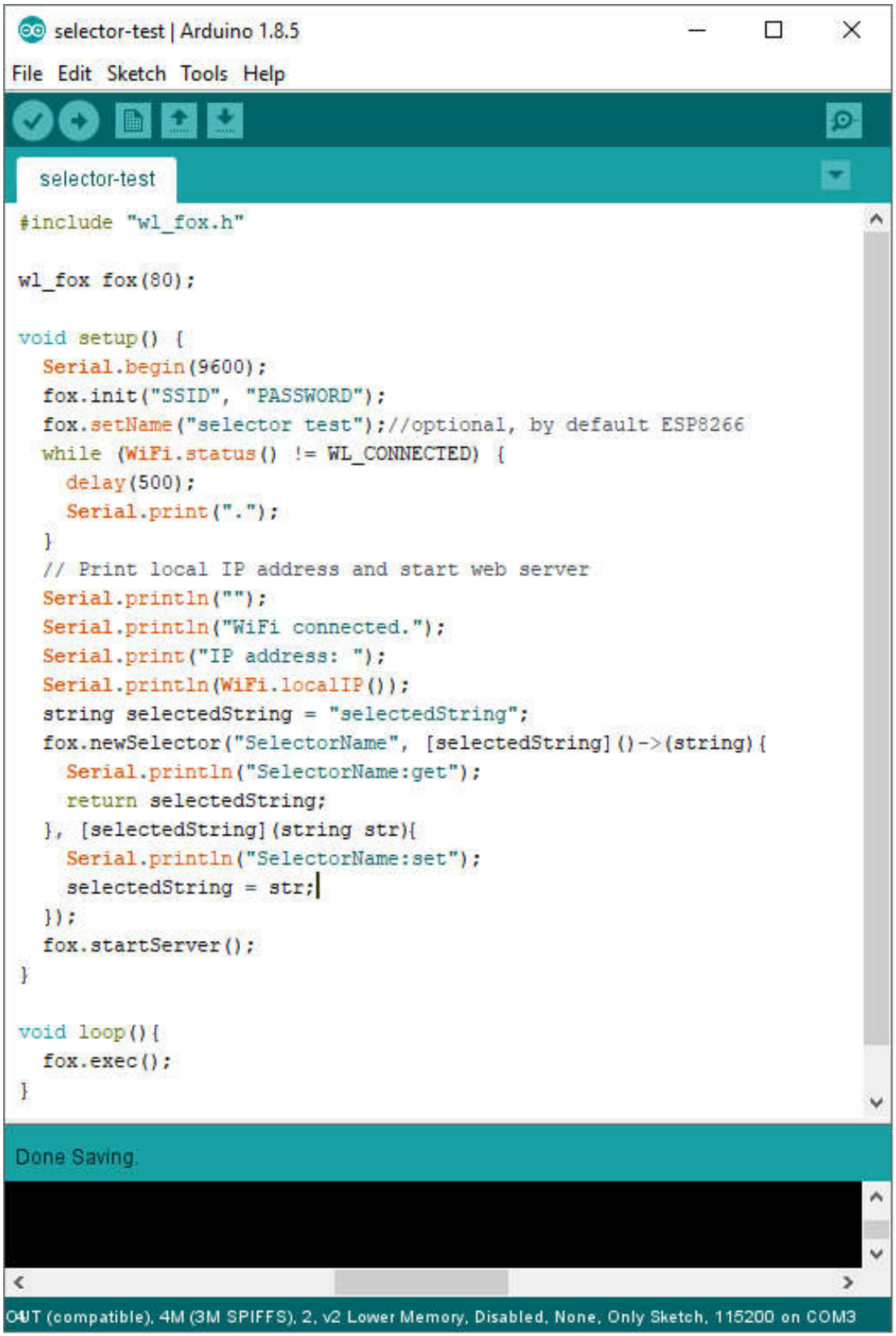

Рисунок 5.13 – Програма для тестування елемента типу «selector»

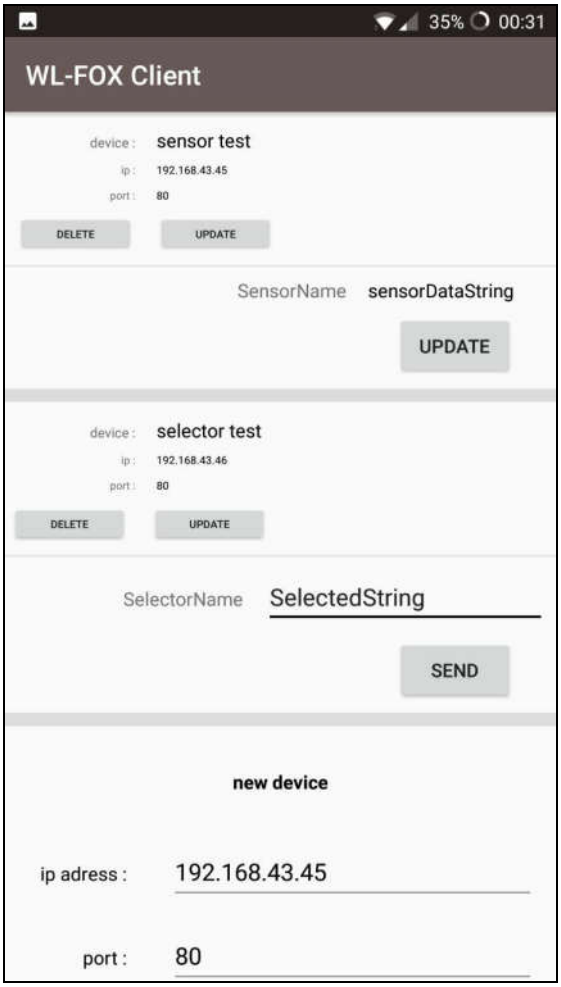

Рисунок 5.14 – Інтерфейс мобільного додатку після оновлення другого пристрою

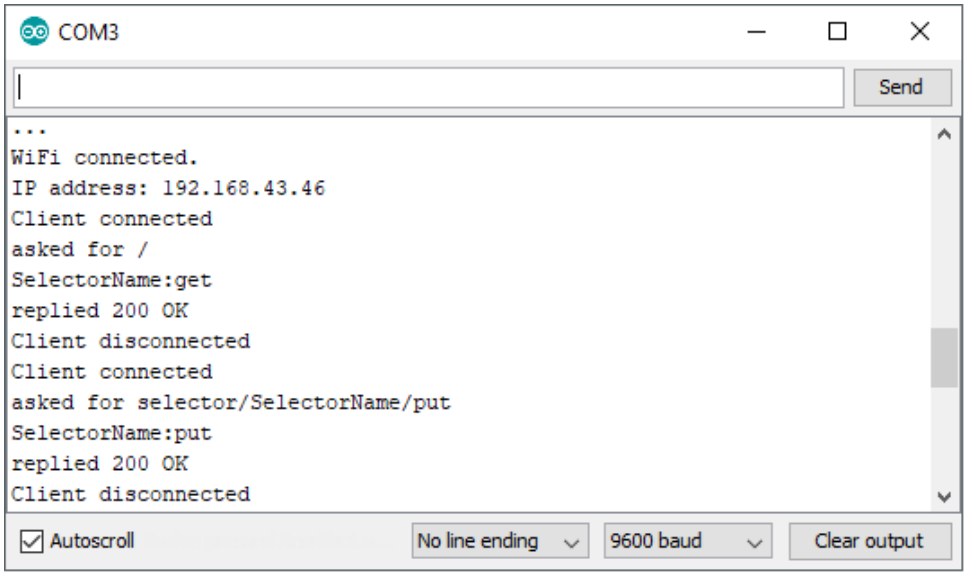

Рисунок 5.15 – Логи другого пристрою для тестування елемента «selector»

Для перевірки можливості видалення пристроїв в мобільному додатку було видалено другий пристрій (рис. 5.16).

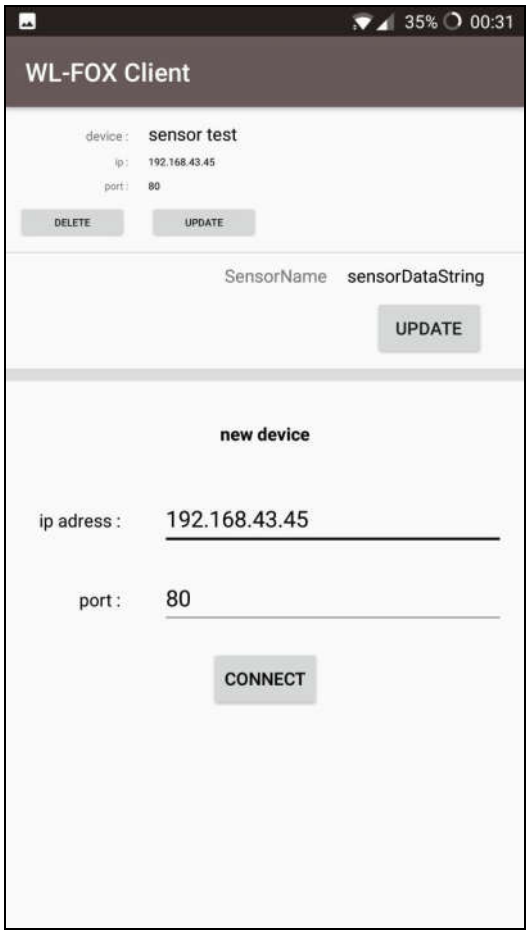

Рисунок 5.16 – Інтерфейс мобільного додатку після видалення другого пристрою

Для перевірки працездатності всіх пристроїв одночасно було написано тестову программу (рис. 5.17) і перезаписано програму першого пристрою. Після оновлення пристрою в мобільному додатку (рис. 5.18) пристрій успішно приймає та обробляє всі типи запитів (рис. 5.19).

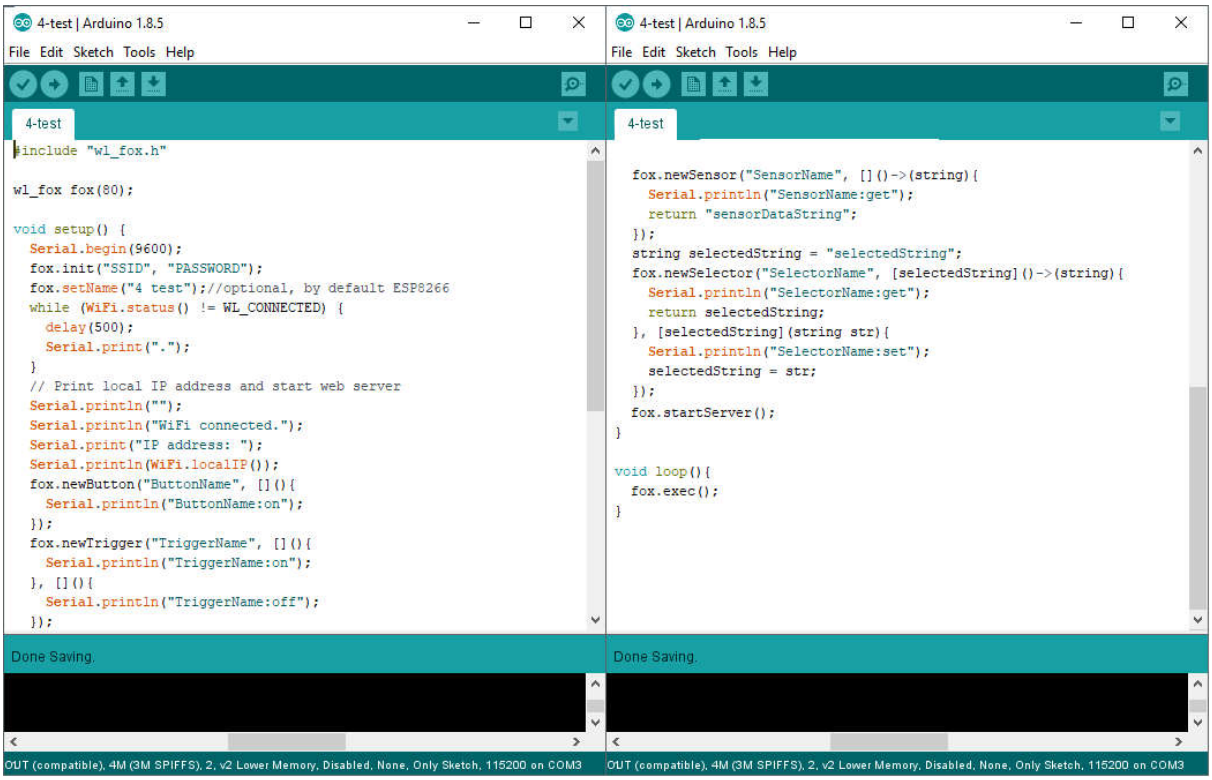

Рисунок 5.17 – Програма для тестування елементів типів «button», «trigger», «sensor» і «selector»

Рисунок 5.18 – Інтерфейс мобільного додатку після другого оновлення першого пристрою

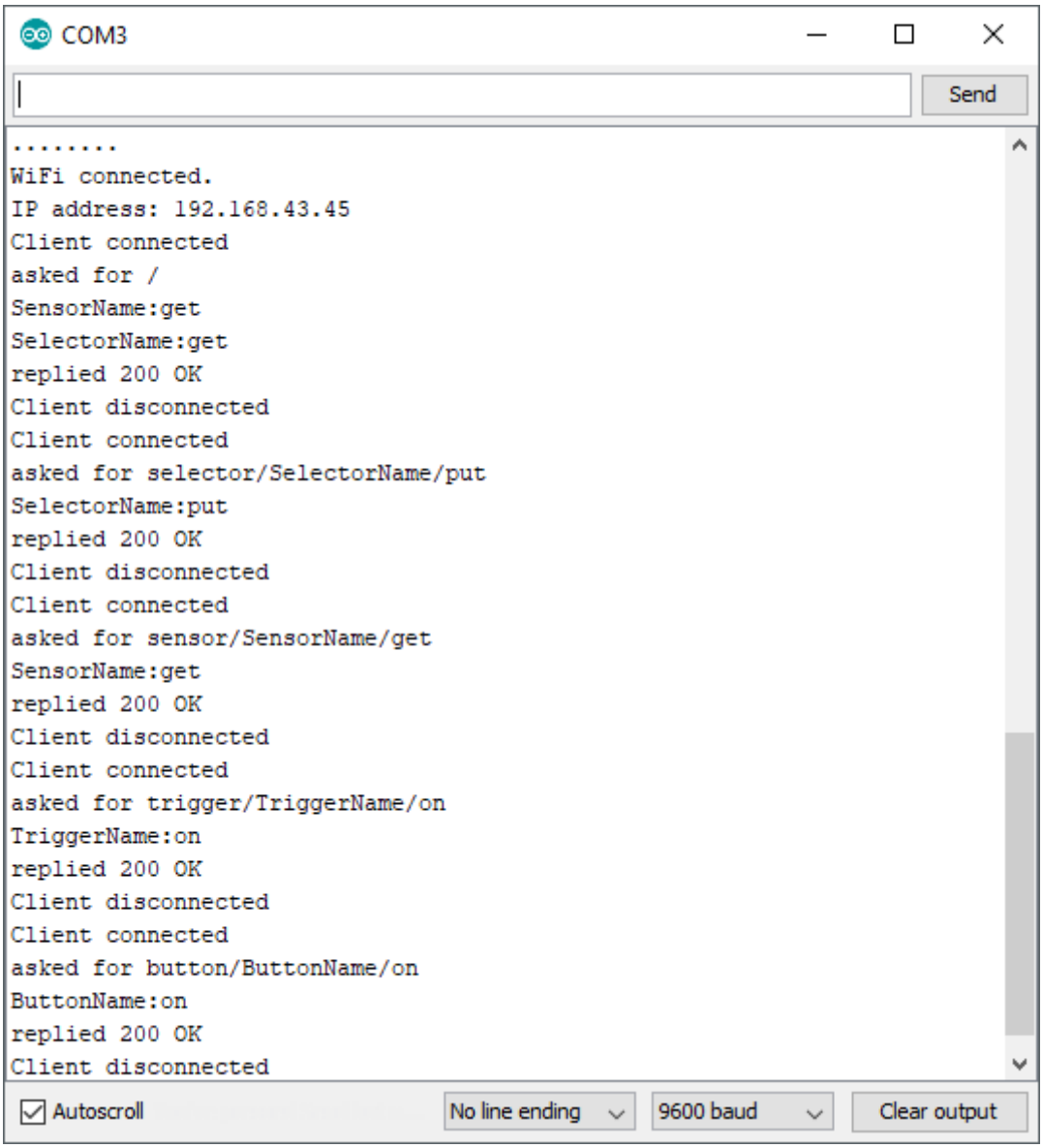

Рисунок 5.19 – Логи першого пристрою для тестування елементів типів «button», «trigger», «sensor» і «selector»

Для перевірки правильності роботи пристроїв при наявності декількох елементів одного типу було написано тестову програмку, що містить два тригера (рис. 5.20) і перезаписано програму першого пристрою. Після оновлення пристрою в мобільному додатку (рис. 5.21) пристрій успішно приймає та обробляє всі типи запитів (рис. 5.22).

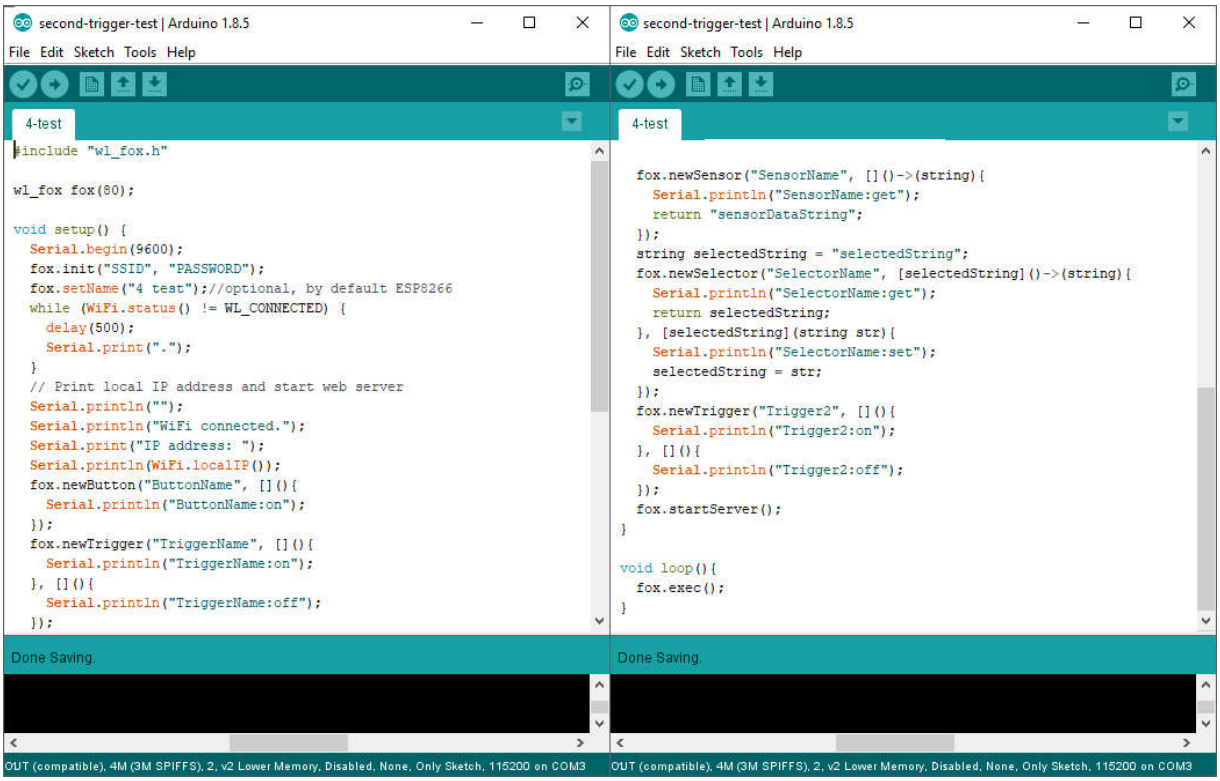

Рисунок 5.20 – Програма для тестування елементів типів «button», «sensor», «selector» і двох елементів типу «trigger»

Рисунок 5.21 – Інтерфейс мобільного додатку після третього оновлення першого пристрою

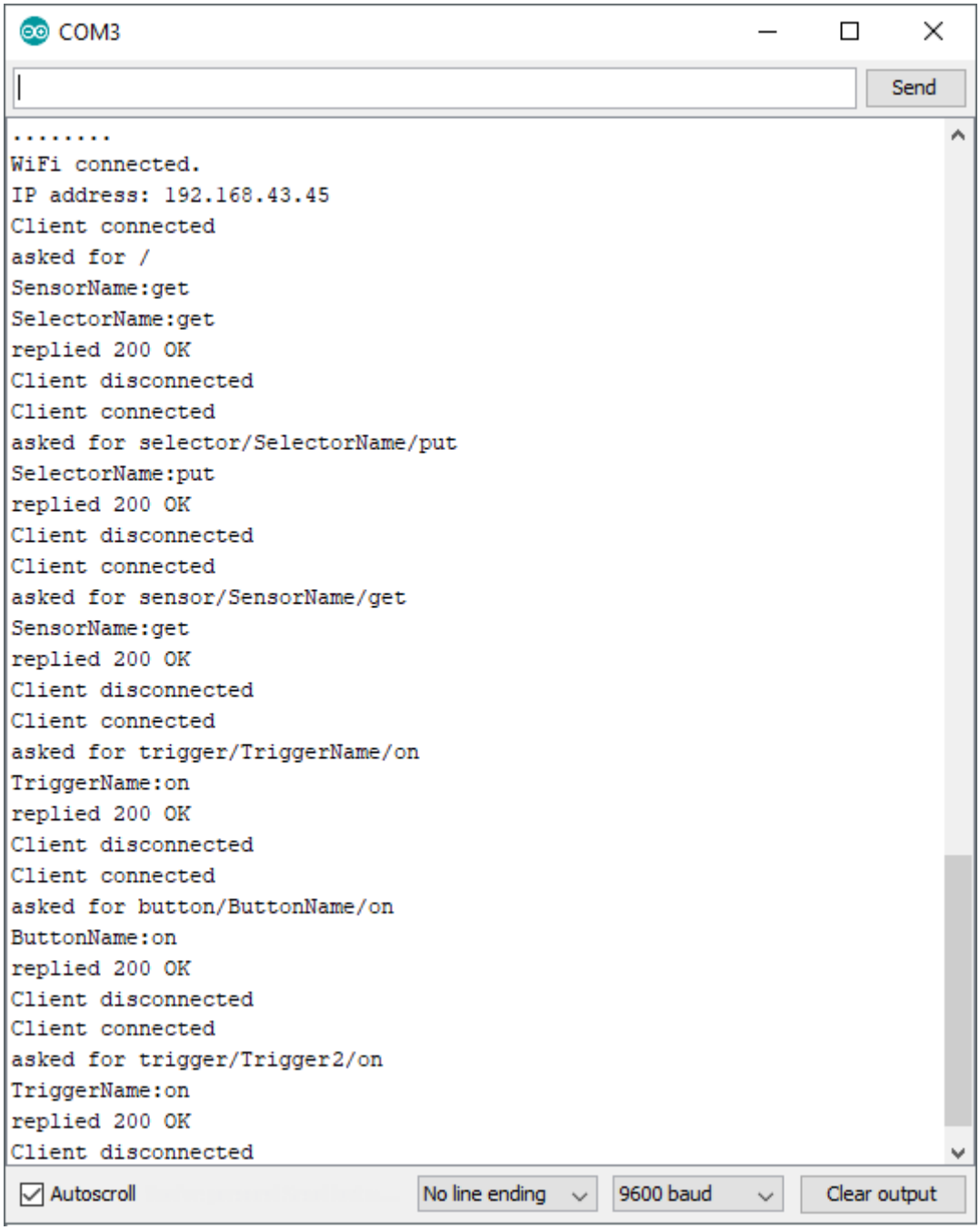

Рисунок 5.22 – Логи першого пристрою для тестування елементів типів «button», «sensor», «selector» і двох елементів типу «trigger»

Для перевірки правильності обробки помилок мобільним додатком, було зроблено спробу підключити неіснуючий пристрій до мобільного додатка, на що було успішно створено повідомлення про помилку (рис. 5.23).

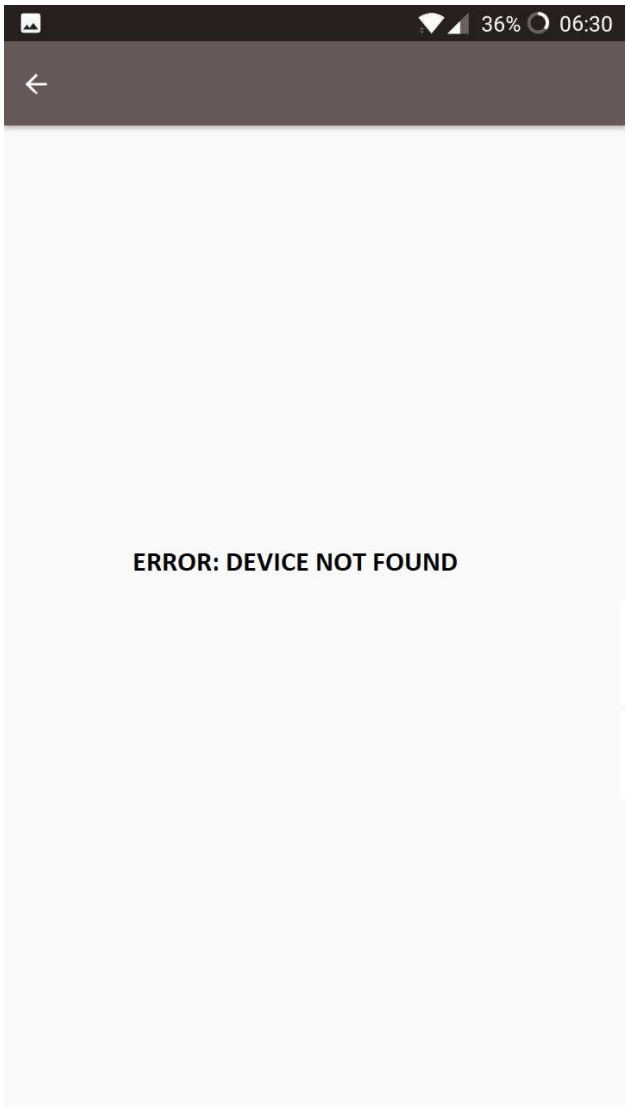

Рисунок 5.23 – Інтерфейс мобільного додатку після спроби помилкового підключення

ВИСНОВКИ

Під час роботи над даним дипломним проектом був розроблений мобільний додаток для управління пристроями системи «Розумний дім», що дозволяє розробникам автоматизувати розробку мобільного інтерфейсу для пристроїв системи «Розумний дім» за рахунок використання спеціально розробленого інтерфейсу комунікації.

В ході виконання даного дипломного проекту було проаналізовано проблему сумісності пристроїв систем «Розумний дім» різних розробників.

Дослідження першого розділу показали, що проблема сумісності пристроїв систем «Розумний дім» є дуже актуальною на сьогодні. Особливо виділяється ця проблема в малих компаніях та некомерційних проектах, де часто для кожного типу пристроїв, а інколи і для кожного пристрою одного типу, необхідно встановлювати окремий мобільний додаток для управління. Це може призводити до труднощів при пошуку необхідного додатку для управління серед встановлених.

В роботі проаналізовані різні типи пристроїв систем «Розумний дім», на основі чого розроблено інтерфейс комунікацій, який охоплює основні сценарії роботи з пристроями. Цей інтерфейс також було розроблено з урахуванням можливості спрощеного модифікування в майбутньому.

Для спрощення роботи з розробленим інтерфейсом комунікації було розроблено бібліотеку, що реалізує наведений інтерфейс для програмуванні мікросхеми ESP8266 в середовищі розробки Arduino IDE. Структура бібліотеки дає можливість адаптувати її до різних мікросхем та середовищ розробки вносячи мінімальні правки.

Розроблений мобільний додаток для управління пристроями «Розумний дім» дозволяє підключатись до необмеженої кількості

пристроїв, що мають довільну кількість елементів управління. Структура мобільного додатку спрощує його модифікацію в майбутньому.

Варто зазначити, що даний проект охоплює тільки базові функції системи «Розумний дім», проте він вже готовий до впровадження в будинках і підприємствах. Завдяки своїй структурі розроблений мобільний додаток також надає можливість швидкої модифікації та адаптації до потреб користувача і тому має потенціал необмеженого росту, що дозволить в майбутньому охопити не тільки базові функції систем «Розумний дім», але й спеціалізовані та більш складні.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Amazon Alexa Features: Smart Home [Електронний ресурс] – Режим доступу до ресурсу: https://www.amazon.com/b?ie=UTF8&node=17934679011
2. Connected Home Devices & Entertainment Systems - Google Store [Електронний ресурс] – Режим доступу до ресурсу: https://store.google.com/ca/category/connected_home
3. Xiaomi Mi Smart Sensor Set [Електронний ресурс] – Режим доступу до ресурсу: https://www.mi.com/global/mi-smart-sensor-set/
4. ESP8266 Community Forum [Електронний ресурс] – Режим доступу до ресурсу: https://www.esp8266.com
5. Blynk IoT platform [Електронний ресурс] – Режим доступу до ресурсу: https://blynk.io
6. Arduino [Електронний ресурс] – Режим доступу до ресурсу: https://www.arduino.cc
7. Arduino ESP8266 web server project [Електронний ресурс] – Режим доступу до ресурсу: https://create.arduino.cc/projecthub/harshmangukiya/create-esp8266-web-server-9c32ac
8. Mobile Operating System Market Share Worldwide | StatCounter Global Stats [Електронний ресурс] – Режим доступу до ресурсу: http://gs.statcounter.com/os-market-share/mobile/worldwide/2018
9. Jacobson D. APIs: A Strategy Guide / D. Jacobson, G. Brail, D. Woods., 2011. – 150 с. – (1st Edition). – (Creating Channels with Application Programming Interfaces).
10. HTTP: The Definitive Guide / [D. Gourley, B. Totty, M. Sayer та ін.]., 2002. – 658 с. – (1 edition). – (Definitive Guides).

11. Bassett L. Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON / Lindsay Bassett., 2015. – 126 с. – (1st Edition).
12. Микитенко С. С. ПІДХІД ДО КОМУНІКАЦІЇ ПРИСТРОЇВ СИСТЕМИ РОЗУМНОГО ДОМУ / С. С. Микитенко, М. М. Орлова. // ІНТЕГРОВАНІ ІНТЕЛЕКТУАЛЬНІ РОБОТОТЕХНІЧНІ КОМПЛЕКСИ. – 2019. – №12. – С. 245–246.
13. Arduino IDE [Електронний ресурс] – Режим доступу до ресурсу: https://www.arduino.cc/en/Main/Software.
14. Bloch J. Effective Java / Joshua Bloch., 2018. – 412 с. – (3rd Edition).
15. Sciore E. Java Program Design: Principles, Polymorphism, and Patterns / Edward Sciore., 2018. – 476 с. – (1st edition).
16. Phillips B. Android Programming: The Big Nerd Ranch Guide / B. Phillips, C. Stewart, K. Marsicano., 2017. – 9998 с. – (3rd Edition). – (Big Nerd Ranch Guides).
17. Jayakumar M. Nodemcu dev kit using Arduino IDE: Get started with ESP8266 / Magesh Jayakumar., 2015. – 88 с. – (1.0 edition).